Final Report

Project Aim

The aim of this web project is to develop a user-friendly Recipe Finder and Nutrition Analysis web application. This application will empower users to discover, search for, and explore a vast database of recipes and consult detailed nutritional information for each recipe. The project focuses on creating a seamless and intuitive user experience to help individuals make informed and healthy food choices.

Description

EAT EASIER is web project designed to help people explore, create, and enjoy their meals. This comprehensive platform seamlessly integrates recipe discovery and nutrition analysis, empowering users to make informed dietary choices while satisfying their taste buds.

1. **Recipe Finder:**

- Discover a vast collection of recipes spanning various cuisines, dietary preferences, and meal types.
- Search by ingredients ensuring a personalized culinary experience.

2. **Nutrition Analysis:**

- Gain insight into the nutritional content of each recipe, including calorie count, macronutrients (carbohydrates, proteins, fats), vitamins, and minerals.
- Set dietary goals and track your daily intake, promoting healthier eating habits.

Functional Requirements

Recipe Management:

- Users can search database of recipes.
- Users can filter recipes based on criteria such as cuisine, ingredients, and dietary restrictions.

Nutrition Analysis:

- The system calculates and displays nutritional information for each recipe, including calories, macronutrients, vitamins, and minerals.

- Users can view a summary of the daily nutritional intake based on the selected recipes.

Meal Planning:
- Users can create meal plans by selecting recipes for breakfast, lunch, dinner, and snacks.

API Documentation:
- Comprehensive API documentation for developers who may want to integrate the service into other applications.

CI/CD (Continuous Integration and Continuous Deployment):
- Implement CI/CD pipelines to automate testing, deployment, and updates.
- Frequent updates and bug fixes to ensure a smooth user experience.

Hosting:
- Hosting on an AWS server.

Testing:
- Comprehensive test cases to ensure the application's functionality, security, and performance.
- Regular testing and quality assurance processes to identify and fix issues.

Non-Functional Requirements

Performance:
- The website load quickly, even during peak usage times.
- The API calls have low latency for a smooth user experience.

Scalability:
- The system was designed to handle a growing user base and recipe database.

Reliability:
- The system is available and reliable, with minimal downtime.

Usability:
- The user interface is intuitive and user-friendly, following material design principles.

User Stories

As a user interested in cooking and nutrition, I want to:

Search for Recipes.
So that: Easily discover a variety of recipes from different cuisines and dietary preferences and search for recipes by ingredients, making it convenient to find dishes based on what I have in my kitchen.

View Recipe Details:
So that: See high-quality images to visually guide me through the cooking process.

Analyze Nutritional Information:
So that: Gain insight into the nutritional content of each recipe, including calorie count, macronutrients (carbohydrates, proteins, fats), vitamins, and minerals. And, set my dietary goals and track my daily nutritional intake for a healthier lifestyle.

Access API Documentation:
So that: Find comprehensive API documentation for developers who may want to integrate the service into other applications.

Experience a Reliable and Fast Platform:
So that: Use a platform that loads quickly and remains responsive even during peak usage times.

Use an Intuitive User Interface:
So that: Navigate a user-friendly and intuitive user interface following material design principles.

Individual's role and responsibilities

Diana
Project Description
Mock-up design

UI Interface

API calls

Hosting (on an AWS server)

API Documentation & How to use

Using CI/CD tools

Test cases

Final report

Adriano

Project Description

Mock-up design

UI Interface

API calls

Hosting (on an AWS server)

API Documentation & How to use

Using CI/CD tools

Test cases